Ciencia e Ingeniería
Revista Interdisciplinar de Estudios en Ciencias Básicas e Ingenierías.
Año 2017, Julio- Diciembre, Vol. (4) N° (2) ISSN 2389-9484.
Universidad de La Guajira, Facultades de Ciencias Básicas y Aplicadas e Ingeniería.
La Guajira-Colombia.
Revista en Línea http://revistas.uniguajira.edu.co/index.php/cei





ARQUITECTURA BASADA EN ARDUINO PARA EL CONTROL DE UN DISPOSITIVO ROBOTICO A TRAVES DE LA RED CELULAR CON ANDROID

ARDUINO BASED ARCHITECTURE FOR ROBOTIC CONTROL DEVICE THROUGH RED CELL WITH ANDROID

Telecom Yair Rivera Julio, Carlos Gonzales y Andrea Mantilla

Universidad Americana, Universidad de Tunja,

Recibido: Noviembre 25 de 2016 Aceptado: Marzo 20 de 2017

RESUMEN

El avance tecnológico en la robótica sostiene muchos diseños y modelos para las distintas aplicaciones industriales, con todo esto cada una de estas soluciones implican un alto valor comercial debido a las exigencias tecnológicas y de infraestructura, motivo por el cual se hace necesario el desarrollo de una arquitectura open source. La cual basada en Arduino es posible conectarlo a cualquier prototipo robótico y operarlo remotamente a través de la red celular 3g, esta conexión permitiría realizar movimientos programados con ayuda de Android desde un dispositivo celular conectado a la red de datos.

Palabras clave: Arduino; Android; SSC-32.

ABSTRACT

Technological advances in robotics holds many designs and models for various industrial applications, with all that each of these solutions involve a high commercial value due to technological and infrastructural requirements , why develop a becomes necessary open source architecture . Which based on Arduino may be connected to any robotic prototype and operated remotely via 3G cellular network, this connection would allow programmed motions using Android from a mobile device connected to the data network.

Keywords: Arduino, Android , SSC- 32

1. INTRODUCCIÓN

El desarrollo de los sistemas robóticos manipulados en tiempo real, modulares y programables se ha convertido en un área de investigación dentro del marco de la llamada inteligencia artificial, por tal motivo han alcanzado una nueva etapa en su desarrollo con placas programables, en este caso las llamadas Arduino o micro controladores programables, la cual nos permite automatizar procesos industriales de forma flexible y a bajo costo.

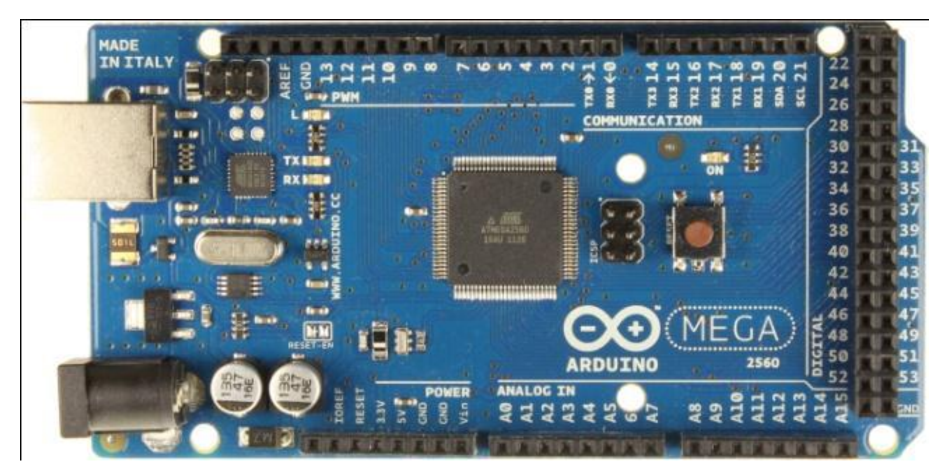
La arquitectura comienza con la programación de una controladora Arduino y un dispositivo móvil los cuales se comunican a través de la red de datos “INTERNET”, la cual permite el intercambio bidireccional de comunicación, todo esto con el objetivo de tener un total control de un dispositivo robótico a partir de un dispositivo móvil, en este caso un teléfono celular.

Las especificaciones de este proyecto consiste en un componente de hardware y otro de software, entre los componentes principales de hardware tenemos una controladora Arduino y un sistema robótico comandado a través de una arquitectura embebida, controladora SSC-32, en cuanto al software este se encuentra distribuido en forma de cliente-servidor, el cliente es el software que corre en el dispositivo móvil y es desde donde parten todas las instrucciones, el servidor es un software programado en la controladora que comanda directamente todas las actividades del robot.

La finalidad de este proyecto es mostrar una arquitectura open source y a bajo costo para programar robots con aplicaciones industriales y que cumplan al mismo tiempo instrucciones generadas a partir de dispositivos móviles (Aroca et al., 2012).

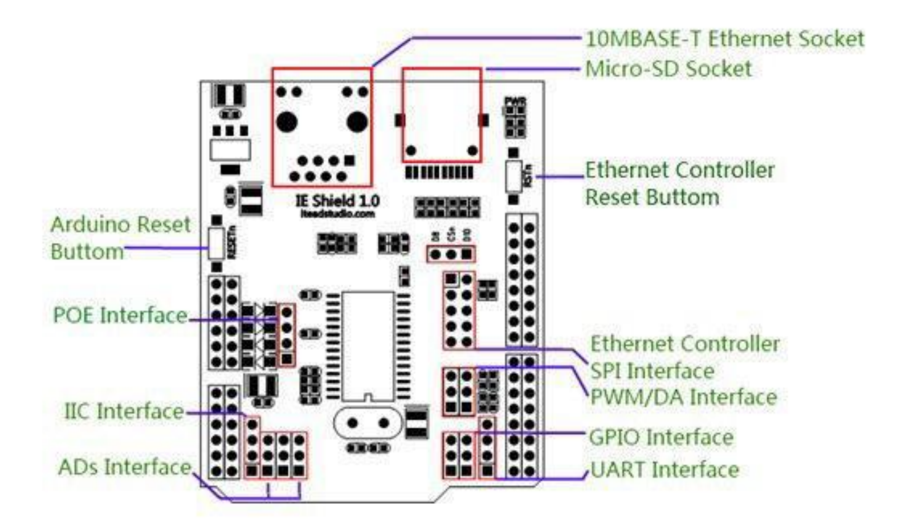
II. ARQUITECTURA DE HARDWARE

Es una placa microcontrolador basada ATmeg1280, Tiene 54 entradas/salidas digitales (de las cuales 14 proporcionan salida PWM), 16 entradas digitales, 4 UARTS (puertos serie por hardware), un cristal oscilador de 16MHz, conexión USB, entrada de corriente, conector ICSP y botón de reset. Contiene todo lo necesario para hacer funcionar el microcontrolador; simplemente conéctalo al ordenador con el cable USB o aliméntalo con un trasformador o batería para empezar, ver siguiente ilustración (Lin y Ye, 2004) (fotografía 1).

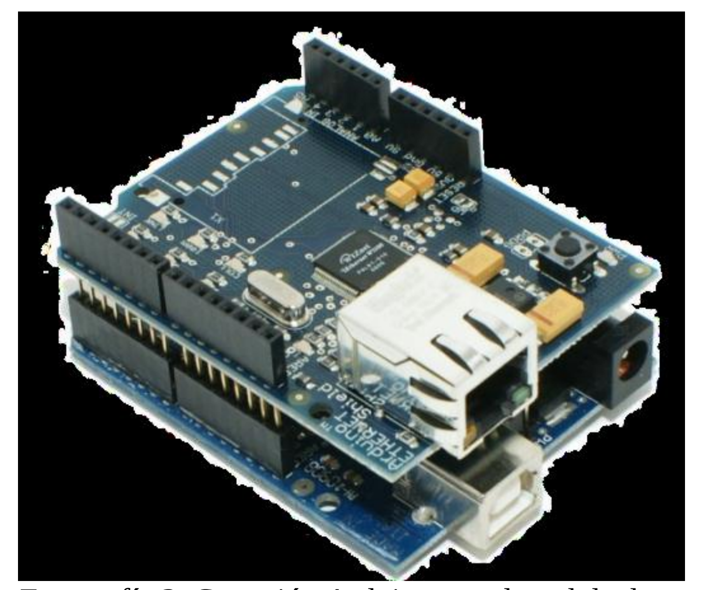


Fotografía 1. Ilustración 1 Arduino Mega 250

Para que esta placa controladora tenga conexión a la red de datos, es necesaria conectarla con una placa modular Ethernet para Arduino, ver Fotografías 2 y 3.



Fotografía 2. Modulo Ethernet para Arduino



Fotografía 3. Conexión Arduino para la red de datos

III. ARQUITECTURA DE SOFTWARE-CLIENTE/SERVIDOR

Previamente se puede programar las instrucciones en un dispositivo móvil conectado a una red de datos IPV4, de tal manera que las instrucciones se pueden registrar de forma remota, inclusive a distancias mayores, el sistema a desarrollar es android 4.0.2, un sistema operativo Open source basado en el kernel de Linux, diseñado para tabletas o teléfonos celulares inteligentes, este sistema fue desarrollado para los estándares abiertos de dispositivos móviles, el sistema fue seleccionado por su estabilidad y rapidez con que se ejecuta las tareas, las herramientas de software tenidas en cuenta para el desarrollo de código fueron para el cliente móvil, IDE Eclipse y el ADT (Android Development Tool) y para la programación del servidor de instrucciones en Arduino (Mohammadi et al., 2012).

El siguientes es el código desarrollado en android el cual envía los paquetes desde el dispositivo móvil

```
IPAddressremote = Udp.remoteIP();
Serial.println(Udp.remotePort());
Udp.read(ReplyBuffer,UDP_TX_PACKET_MAX_SIZE);
Serial.println("Contents:");
Serial.println(packetBuffer);
Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
Udp.write(ReplyBuffer);
Udp.endPacket();
```

IV. SERVIDOR ARDUINO Y CODIFICACION DE INSTRUCCIONES SSC-32

Teniendo en cuenta toda las conexiones necesarias y procedemos a programar, tomamos la controladora Arduino y se programa de tal forma que actué como un servidor UDP/IP4, desde donde se recibirá cada paquete enviada desde la red celular, gracias a la shield Ethernet para Arduino podemos asignarle un numero IP y habilitar un numero de puerto udp para la recepción de los paquetes, ver siguiente código en Android (Sheridan, 1992).

```
bytemac[] = {
0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddressip(192, 168, 1, 177);
char ReplyBuffer[] = "acknowledged";
EthernetUDPUdp;
unsignedintlocalPort = 8888;
Ethernet.begin(mac,ip);
Udp.begin(localPort);
Serial.begin(9600);
```

La controladora Arduino ahora conectada a la red de datos puede capturar cualquier paquete udp y codificarlo, para este laboratorio se trabajó con una controladora SSC-32 de comunicación serial, la cual permite el control de 32 servos, todos conectados directamente con un rango de movimiento 180 con una resolución de 1uS por tiempo, propiedades que permiten manejar movimiento en tiempo real (fotografía 4).

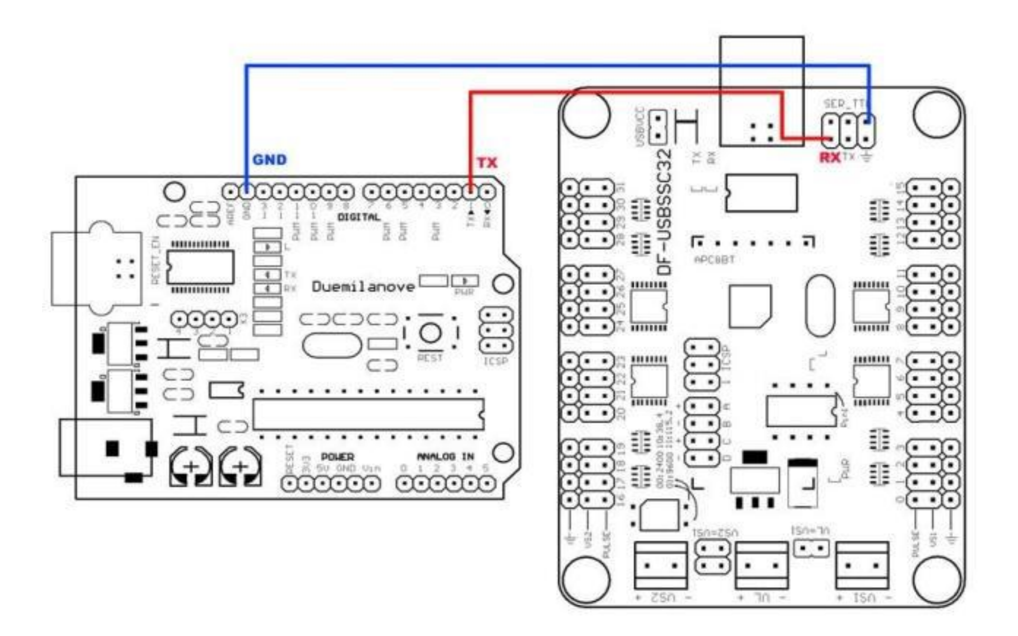


Fotografía 4. Controladora Ssc-32

Una componente moldeada por inyección personalizada y ultra-resistentes, con un torso totalmente articulado y compuesto por 8 x HS-645MG, 3 x HS-475HB / HS-485HB, y 3

x HS-422 servos, Incluye dos 12vdc 50:1.Todo con una excelente tracción diseñado y desarrollado por Lynxmotion, Inc (Pérez, 2011).

Teniendo en cuenta estos dispositivos, procedemos a realizar la interface entre la Arduino y la SSC-32, Ver Fotografía 5.



Fotografía 5. Controladora Ssc-32

La comunicación entre la controladora Arduino y el dispositivo móvil se da de forma bidireccional con una velocidad de 9600 baudios, la controladora puede tener funciones de clientes como de servidor, para este caso puede mandar paquetes UDP a cualquier dispositivo móvil, para esta prueba se le envía al mismo dispositivo móvil cliente un paquete UDP como respuesta (Oldenberg and Ondik, 20011). La variable packet Buffer, es donde se encuentra la información que el dispositivo móvil le envía a la controladora arduino, Esta las codifica y ejecuta las instrucciones necesarias para remitir los bytes hacia la controladora servo SSC-32 en forma serial (Zorlu et al., 2011).

Finalmente después que se ejecutan los movimiento de forma sincronizarse procede a enviar un una confirmación de movimientos ejecutados hacia el dispositivo móvil. Ver el siguiente código (Gundeti, 2012).

```
Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
Udp.write(ReplyBuffer);
Udp.endPacket();
```

Cuando la información enviada desde el dispositivo móvil y esta es recibida en la controladora arduino, se puede determinar que instrucción enviar hacia la controladora servo, y así controlara cada servo en un tiempo determinado y por separado, simplemente se debe tomar el siguiente formato de instrucción:

```
Serial.println("#int servo, int position, int time)
```

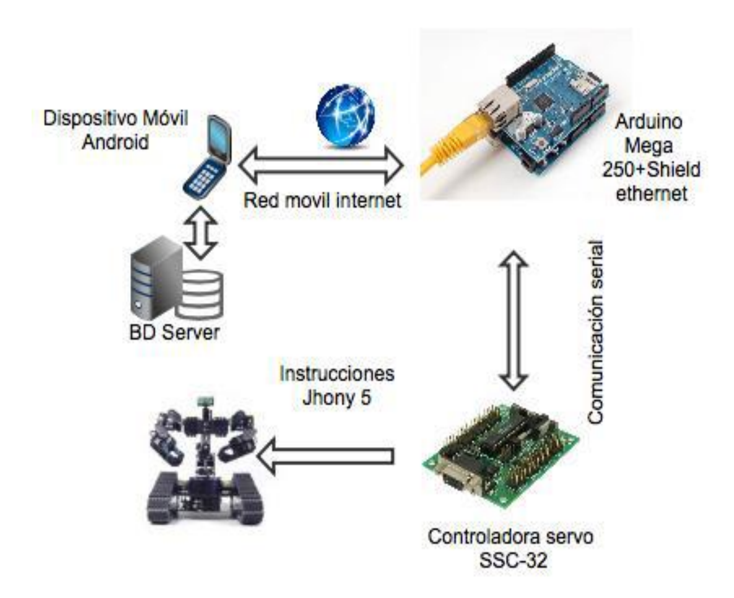
Donde se selecciona el motor servo, con la posición position y una velocidad time.

Ejemplo: se debe articular el motor número 27, el cual es la mano de jhony 5, en una posición 500 con una rapidez de 500 milisegundos

Serial.println("#27 P1000 T500"); //motor mano (H) 1
delay(1000);

V. ARQUITECTURA GENERALIZADA

Se integra la eficiencia de una plataforma móvil con un sistema robótico donde se realizan movimientos en tiempo real, conectados a través de internet para una operación remota, una arquitectura fácil de implementar. Opcionalmente se puede interactuar con una base de datos, en la cual se almacenan la serie de movimientos a ejecutar por el jhony 5.

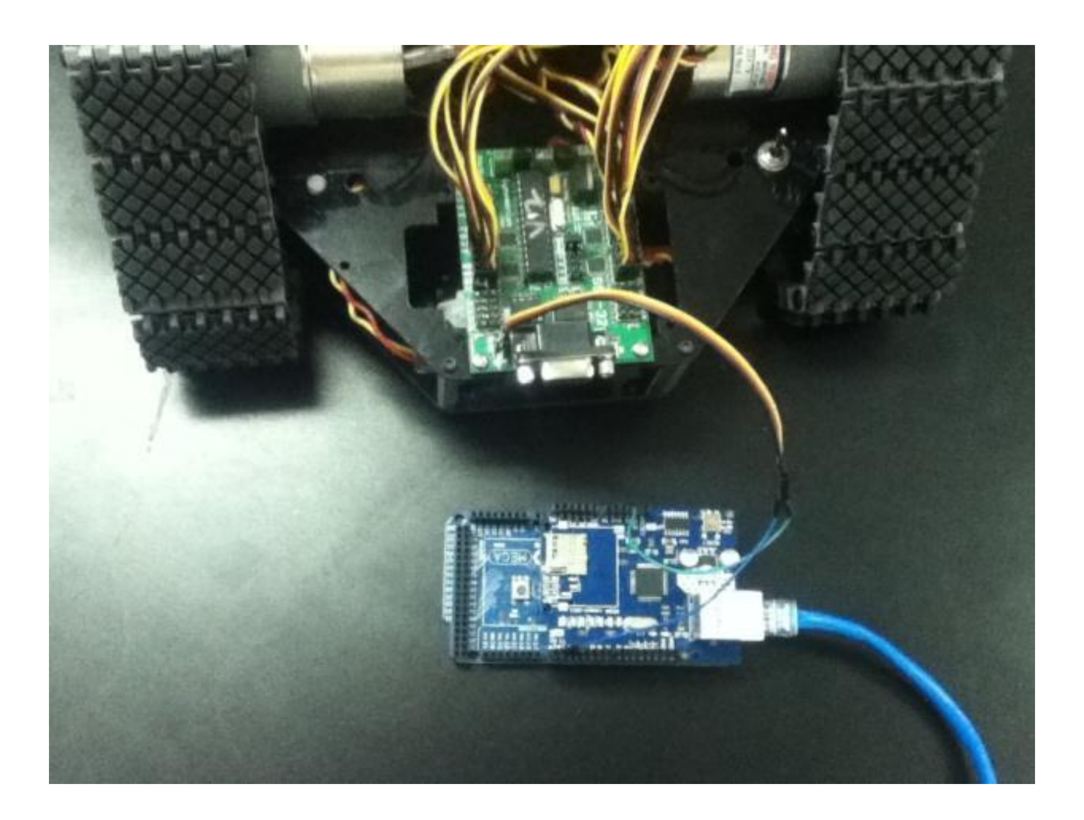


Fotografía 6. Diagrama de la Solución Planteada

La solución se compone de 2 áreas (Girones,2011):

1. una área local o maestra compuesta por el dispositivo móvil, desde de donde interactúan todos los servicios de red a través de la internet.
2. Un área remota o esclava, en la cual se encuentra el dispositivo robótico manipulado remotamente por la SS-32.

La comunicación entre las dos áreas se puede dar full dúplex, ya que el dispositivo robótico también puede enviar paquetes hacia la red de datos a través de la controladora arduino. Ver Fotografía 7.



Fotografía 7. Interface SSC-32-ARDUINO

Se podría mejorar realizándose inalámbricamente.
La comunicación entre la controladora arduino y la red de datos

VI.CONCLUSIONES

Se presentó un diseño embebido basado en una arquitectura open source tanto en hardware como en software, a bajo costo de tal manera que se pueden automatizar tareas en tiempo real y de forma remota, a través de la red de datos, una red que permite un control full dúplex, las razones por la cual se desarrolló la arquitectura son los altos costos de sistemas personalizados los cuales son cerrados y por tal motivos no se dejan personalizar, a menos que el proveedor exija un costo adicional por su modificación.

VII. REFERENCIAS

Aroca, Pe ricles, and onc alves, “Towardssmarter robots withsmartphones,” Robocontrol: Proc. 5th , pp. 1–6, 2012.

Arduino. (2012, Dec. 20). SoftwareSerial Library. Available: http://arduino.cc/en/Reference/SoftwareSerial

Android. (2012, Dec. 6th). Connectivity. Available: http://developer.android.com/guide/topics/connectivity/index.html

Lin and W Ye, “OperatingSystemBattle in theEcosystem of Smartphone Industry,” 2009 International SymposiumonInformationEngineering and Electronic Commerce, no. 2004, pp. 617–621, 2009.

Mohammadi, M. Tavakoli, and H. J. Marquez, “Control of nonlinearteleoperationsystemssubjecttodisturbances and variable time delays,” 20 2 IEEE/RSJ International ConferenceonIntelligent Robots and Systems, pp. 3017–3022, Oct. 2012.

Sheridan, “Telerobotics, utomation, and Human Supervisory Control,” MitPress, 1992.

Goldenberg and M P Ondik, "Da inci robot-assistedtranscervicalexcision of a parapharyngealspace tumor," Journal of RoboticSurgery, vol. 4, no. 3, pp. 197–199, Jul. 2010.

Gundeti, "Pediatric Robotic and ReconstructiveUrology: A Comprehensive uide " Wiley, 2012.

Girone s, El ran Libro de Android. Marcombo, 2011.

Meulen, C. Pettey. (2012, Dec. 16). GartnerSaysWorld- wide Sales of Mobile PhonesDeclined 3 Percent in ThirdQuar- ter of 2012; Smartphone Sales Increased 47 Percent. Available: http://www.gartner.com/newsroom/id/2237315

Perez V. Z, "Desarrollo de una metodolog a de realimentacio n de fuerza y torque en cirug a robo tica card aca m nimamente invasiva," Universidad Pontificia Bolivariana.

Perez, J R Mart nez, J Bustamante, M J Betancur, and O P Torres, "Sistema maestro-esclavo basado en dispositivos mo viles para aplicaciones biome dicas," I Congreso Colombiano de Bioingenier a e Ingenier aBiome dica, 20

Sachs, "Sensor FusiononAndroiddevices: A Revolution in Motion Processing," oogle TechTalk, August 2, 2010

Tsai W., Robot Analysis. Themechanics of serial and parallel manipulators. A WileyIntersciencePublication, 1999, p.

Wakerly. (2012, Dec. 18). Android USB host serial driver libraryfor CDC, FTDI, Arduino and otherdevices. Available: http://code.google.com/p/usb-serial-
for-android/

Zorlu, U. Selek, and H. Kiratli, "Initialresults of fractionatedCyberKniferadiosurgeryfor uveal melanoma ," Journal of neuro- oncology, vol. 94, no. 1, pp. 111–7, Aug. 2009.